

Análisis Comparativo

Carlos Federico Gaona

Facultad Politécnica, Universidad Nacional de Asunción, Paraguay

Resumen Se compara dos metaheurísticas sobre el problema de asignación cuadrática, el Algoritmo Genético NSGA II y la Colonia de Hormigas M3AS. Se utilizan 2 instancias de tamaño 75 utilizados en trabajos similares y disponibles *online*. Se concluyó que M3AS alcanzó un estado de estancamiento prematuro y se presentaron ideas y referencias de como atacar este problema.

Keywords: QAP, M3AS, NSGA, Ant Colony, Genetic Algorithm

1. Introducción

En la actualidad los problemas crecen a un ritmo mucho mayor de que las computadoras puedan manejar, esto exigió nuevas estrategias para resolverlas. Una de ellas es relajando la exigencia de optimalidad, es decir no exigimos una certidumbre absoluta sobre la optimalidad de la solución a presentar. Los Algoritmos Evolutivos y las Colonias de Hormigas son dos familias de metaheurísticas que implementan esta estrategia.

Como era de esperarse, luego de esta transformación surgen nuevos problemas que se vuelven cada vez mas comunes. A continuación presentamos una comparación simple entre 2 algoritmos sobre el problema de la asignación cuadrática.

2. Formulación del Problema

2.1. Problema de la Asignación Cuadrática

El Problema de la Asignación Cuadrática o *Quadratic Assignment Problem* (QAP) es uno de los problemas fundamentales en optimización combinatoria y es considerado uno de los más difíciles de su clase. En la práctica solo se puede resolver de forma óptima instancias de hasta $n \approx 30$ [2].

Dado dos matrices de $n \times n$ $D = \{d_{ij}\}$ y $F = \{f_{ij}\}$ encontrar la permutación ϕ sobre $\{1, 2, \dots, n\}$ tal que se minimize la ecuación 1. Usualmente se aplican mas restricciones como que D y F sean simétricas y no negativas o que d_{ij} sea una distancia, sin embargo estas restricciones no reducen la dificultad del problema.

$$\sum_{i,j} d_{ij} f_{\phi(i)\phi(j)} \quad (1)$$

Se puede probar que QAP generaliza conocidos problemas como el Problema del Viajero o *Traveling Salesman Problem*, *Clique Cover*, etc.

En esta comparación utilizaremos una variación de este problema añadiendo una matriz de flujos más. De esta forma tendremos dos funciones a optimizar constituyendo en un problema bi objetivo.

3. Algoritmo MOEA

Los Algoritmos Evolutivos Multiobjetivos o *Multi-objective Evolutionary Algorithms* (MOEA) son una familia de metaheurísticas para problemas de optimización inspirados en la evolución biológica. Fundamentalmente aplican operaciones sobre una población de soluciones. Una clasificación de los MOEA son los Algoritmos Genéticos (AG) que utilizan 3 operaciones sobre las soluciones, la combinación, la mutación y la selección. La combinación se encarga de buscar mejores soluciones a partir de buenas soluciones. La mutación es una operación de exploración para añadir diversidad a la población. La selección se encarga de decidir cuales soluciones sobreviven a la siguiente generación afectando la probabilidad de que su “información genética” se mantenga disponible.

Una de estas metaheurísticas es el Algoritmo Genético de Ordenamiento No Dominado o *Non-dominated Sorting Genetic Algorithm II* (NSGA II)[3]. El NSGA II determina un operador de selección sobre Conjuntos Pareto con el objetivo de discriminar las soluciones según su aptitud individual y la extensión del conjunto Pareto.

4. Algoritmo MOACO

Los Algoritmos de Optimización Multiobjetivos por Colonia de Hormigas o *Multi-objective Ant Colony Optimization* (MOACO) son también una familia de metaheurísticas para problemas de optimización inspirados en el comportamiento de las hormigas en busca de alimento. El algoritmo ocurre sobre un grafo por donde un conjunto de agentes u hormigas exploran el espacio solución dejando un rastro de feromonas en función de la aptitud de la solución que representan guiando a las demás hormigas. Los problemas que se plantean sobre grafos son particularmente triviales de aplicar, como por ejemplo el TSP y el QAP.

De las múltiples variaciones de la idea, surge el Sistema de Hormigas Min Max Multiobjetivo o *Multi-objective Min Max Ant System* (M3AS)[4] el cual generaliza para múltiples objetivos el *Min Max Ant System*. Se caracteriza por (1) permitir que solo las soluciones no dominadas modifiquen el rastro de feromonas, (2) acotar las modificaciones sobre el rastro mediante τ_{MIN} y τ_{MAX} que están en función a la aptitud de la solución y (3) el rastro se inicializa con valores arbitrariamente grandes para inducir un lapso de exploración al inicio.

5. Resultados Experimentales

Todo el código utilizado está disponible en el repositorio “Agregar!” con detalles sobre su ejecución.

Se utilizó un el siguiente hardware:

Procesador Intel Core i5 6-200U 2.3GHz

Memoria 8GB DDR4 disponibles. Aunque no se utilizó mas de 300MB.

Y en software:

Lenguaje Java 8

Compilador OpenJDK 1.8.0_144

Maquina Virtual OpenJDK 64-bits build 25.144-b01

Bibliotecas MOEAFramework 2.12

La biblioteca MOEAFramework[6] se utilizó para los experimentos con NSGA II. Mientras que M3AS fue implementado.

Como ambos algoritmos son iterativos, utilizamos como criterio de parada una cantidad máxima de iteraciones/generaciones. Esto ayuda a evitar los problemas asociados a los distintos tiempos que ambos algoritmos tardan en completar una iteración.

Para ambas instancias y ambos algoritmos se utilizaron 5000 iteraciones.

5.1. Métricas

Se utilizaron 4 métricas en total para comparar los resultados. Primeramente aclaramos la nomenclatura, la distancia euclidanea se indica con $d(a, b)$, Y_{true} es el frente pareto final construido por la combinación de todos los frentes obtenidos a lo largo del experimento tanto con NSGA II y con M3AS.

Las métricas M1, M2 y M3 son sensibles a los valores de las funciones objetivo, por lo tanto sus valores deben ser interpretados teniendo en mente esto. Para las instancias con las que cuales trabajamos, los objetivos toman valores desde 8000000 a 15000000, aproximadamente.

La primera métrica M1 está definido según la ecuación 2 e indica la “similitud” entre los frentes Y y Y_{true} . Es nonegativo y $M1(Y, X) = 0$ si y solo si $Y = X$. Menores valores indican un mejor rendimiento.

$$M1(Y, Y_{true}) := \frac{\sum_{y \in Y} \min_{y' \in Y_{true}} \{d(y, y')\}}{|Y|} \quad (2)$$

La métrica M2 según 3 indica la “distribución” de las soluciones en el frente, un buen frente tiene sus soluciones uniformemente distribuídas. Es claramente nonegativo y está acotado por $|Y|$ cuando $\sigma = 0$. σ es calculado como el 10 % de la máxima distancia entre dos puntos de Y [5].

$$M2(Y, \sigma) := \frac{\sum_{y \in Y} |\{x \in Y : d(x, y) > \sigma\}|}{|Y| - 1} \quad (3)$$

M3 mide la “extensión” del frente como se ve en la ecuación 4, un buen frente tendrá soluciones muy especializadas en un objetivo. Indicamos con y_i como el valor del i-ésimo objetivo de la solución y para $i = 1, 2, \dots, k$.

$$M3(Y) := \sqrt{\sum_{i=1}^k \max_{x, y \in Y} \{d(x_i, y_i)\}} \quad (4)$$

El error está definido en la ecuación 5 e indica el porcentaje de soluciones que Y comparte con Y_{true} . Es nonegativo y el máximo valor es $|Y_{true}|$. Como Y_{true} es el mejor frente que encontramos entonces mayores valores son preferidos. Utilizamos el Corchete de Iverson, $[P] = 1$ si P es verdadero y 0 en caso contrario.

$$E(Y, Y_{true}) := \frac{\sum_{y \in Y} [y \in Y_{true}]}{|Y_{true}|} \quad (5)$$

6. Conclusiones

En el cuadro 1 y en la figura 1 observamos los resultados de la instancia **qapUni.75.0.1.qap**, en el cuadro 2 y en la figura 2 los de la instancia **qapUni.75.p75.1.qap**.

A partir de las figuras 1 y 2 es evidente que el frente producido por MOEA es dominante. Esto tambien queda en evidencia al analizar los cuadros donde MOEA gana en todas las métricas.

En M1, MOEA obtuvo 0 por que su frente fue efectivamente el frente pareto final Y_{true} mientras que MOACO recibió un valor excesivo debido a los valores que sus objetivos toman (esto se puede apreciar en las figuras).

En M2, MOEA volvió a obtener una calificación significativamente superior. Sin embargo, esta vez fue debido a la cantidad de soluciones que su frente alverga. Se debería analizar el comportamiento de M2 según como varía σ . En ambas figuras observamos cierta escasez de soluciones cerca del extremo hacia el objetivo 1 (eje horizontal) y una aglomeración hacia el objetivo 2. MOACO sufrió un valor bajo debido a la baja cantidad de soluciones. Aunque ambos algoritmos hayan corrido una misma cantida de iteraciones, MOACO contó con 75 hormigas por iteración mientras que MOEA con 100 individuos por iteración.

En M3, los algoritmos estan mas parejos. Aunque MOACO haya producido menos soluciones ha logrado expandirse casi tanto como MOEA. Además los valores son muy similares entre instancias.

En Error el resultado era el esperado luego de ver las figuras. Como Y_{true} es el frente MOEA, este posee la calificación perfecta.

Aunque las diferencias parezcan importantes si nos detenemos a analizar con más cuidado podemos observar que en el objetivo 1 (horizontal) solo hay un 5.8 % y un 6.6 % aproximadamente de mejora en en frente MOEA respecto al frente MOACO. Para el objetivo 2 tenemos un 1.5 % y un 5.8 %, aproximadamente.

Al analizar el *log* de la evolución de los frentes, se llega a la conclusión de que MOACO había llegado a converger muy antes de tiempo desaprovechando el resto de las iteraciones. Aunque el factor de evaporación era bajo, 0.02, la reducción de las feromonas es exponencial lo cual induce a una mayor velocidad de estancamiento.

Utilizamos $\alpha = 1$ y $\beta = 2$, esto según [7], lo cual puede haber sido contraproducente debido a que se dio mucha mas importancia a la heuristica de distancias antes que a la información de feromonas. Es posible que en estas instancias esta heuristica sea de poca ayuda.

Además como se había notado, la cantidad de hormigas en MOACO fue de 75 mientras que MOEA contó con 100 individuos por iteración. Esto también es un punto de desventaja en contra de MOACO, sin embargo no creemos que sea el más significativo.

En conclusión, el temprano estancamiento de MOACO impidió al algoritmo a aprovechar al máximo el tiempo disponible al no **explorar** mas exhaustivamente. Se podría modificar el criterio de evaporación o restaurar las feromonas cuando se detecte el estancamiento (en [7] se detalla un criterio) para volver a explorar. Fué muy importante normalizar los valores de los objetivos al actualizar las feromonas debido a que sus escalas pueden dominar los valores normales de feromonas.

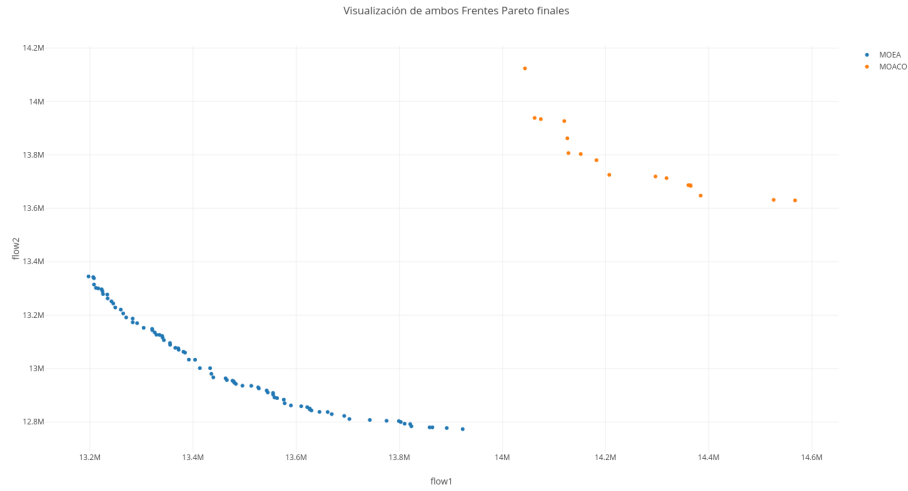


Figura 1. Frentes Pareto producidos por ambos algoritmos para **qapUni.75.0.1.qap**. En el eje horizontal medimos el objetivo 1 mientras que el en eje vertical el objetivo 2.

	M1	M2	M3	Error
MOEA	0.0	15.827	1354.501	1.0
MOACO	1040184.883	4.312	1199.919	0.0

Cuadro 1. Resultados finales sobre la instancia **qapUni.75.0.1.qap**.

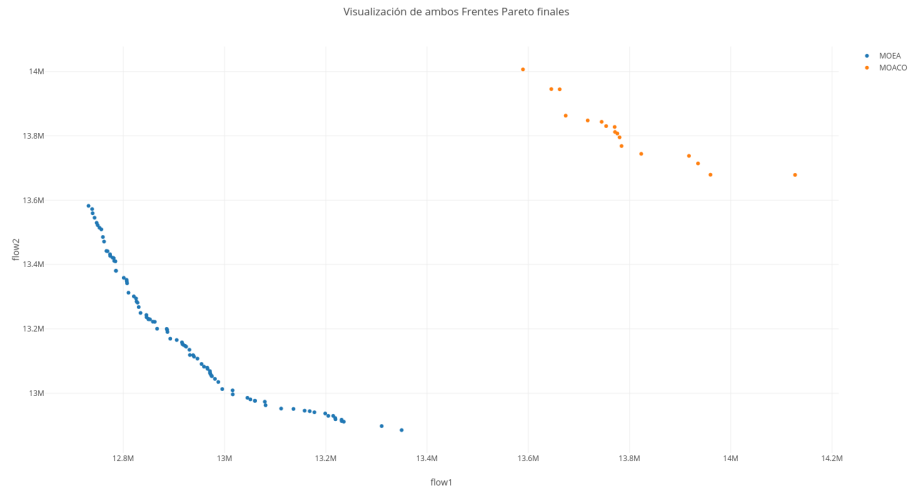


Figura 2. Frentes Pareto producidos por ambos algoritmos para **qapUni.75.p75.1.qap**. En el eje horizontal medimos el objetivo 1 mientras que el en eje vertical el objetivo 2.

	M1	M2	M3	Error
MOEA	0.0	17.239	1363.981	1.0
MOACO	1010546.344	6.188	1106.313	0.0

Cuadro 2. Resultados finales sobre la instancia **qapUni.75.p75.1.qap**.

Referencias

1. Ali Safari Mamaghani, Mohammad Reza Meybodi: Solving the Quadratic Assignment Problem with the Modified Hybrid PSO Algorithm
2. Refael Hassin, Asaf Levin, Maxim Sviridenko: Approximating the minimum quadratic assignment problems
3. K. Deb ; A. Pratap ; S. Agarwal ; T. Meyarivan: A fast and elitist multiobjective genetic algorithm: NSGA-II
4. Pinto, Diego; Barán, Benjamín: Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach (2005)
5. Paciello, Julio; Martínez, Hector; Lezcano, Christian; Barán, Benjamín: Algoritmos de Optimización multi-objetivos basados en colonias de hormigas
6. Hadka, David. "MOEA Framework-A Free and Open Source Java Framework for Multiobjective Optimization."(2012).
7. Thomas Stützle; Holger H. Hoos: MAX-MIN Ant System (2000)